

10/630,525

EIC/STIC search

Set	Items	Description
S1	220490	REWARD? ? OR ENCOURAGEMENT? ? OR REINFORCEMENT? ?
S2	15241772	LEVEL? ? OR DEGREE? ? OR STRENGTH? ? OR MEASUR????? OR EXT- ENT
S3	14274547	VALUE? ? OR QUANTI????????? OR HOW()MUCH OR AMOUNT? ? OR ME- TRIC? ? OR MEASUREMENT? ? OR CALCULATION? ? OR COMPUTATION? ? OR VALUATION? ? OR HEURISTIC? ?
S4	3678019	SUCCESS??? OR ACCOMPLISH???? OR ACHIEVE???? OR MASTERY OR - PROGRESS
S5	1651113	MEMORY? ? OR STORAGE
S6	15039807	MANAG????? OR ADMINISTRAT??? OR MGMT OR MGT OR SUPERVIS??? OR ADJUST???? OR REGULAT??? OR CONTROL???? OR TRACK??? OR MON- ITOR??? OR OVERS?????
S7	211	S1 AND (S2 OR S3) AND S4 AND S5 AND S6
S8	132	S7 AND (PY<2002 OR PD<20010731)
S9	98	RD (unique items)
S10	267	(S1 AND S4 AND S5 AND S6) NOT S9
S11	130	S10 AND (PY<2002 OR PD<20010731)
S12	104	RD (unique items)
S13	63	(AU=(ANDREASSON E? OR ANDREASSON, E?)) NOT (S9 OR S12)
S14	1	S13 AND S1

? show files

File 2:INSPEC 1898-2006/Sep W1
(c) 2006 Institution of Electrical Engineers

File 6:NTIS 1964-2006/Sep W1
(c) 2006 NTIS, Intl Cpyrght All Rights Res

File 8:Ei Compendex(R) 1970-2006/Sep W1
(c) 2006 Elsevier Eng. Info. Inc.

File 34:SciSearch(R) Cited Ref Sci 1990-2006/Sep W1
(c) 2006 The Thomson Corp

File 35:Dissertation Abs Online 1861-2006/Aug
(c) 2006 ProQuest Info&Learning

File 56:Computer and Information Systems Abstracts 1966-2006/Aug
(c) 2006 CSA.

File 57:Electronics & Communications Abstracts 1966-2006/Aug
(c) 2006 CSA.

File 60:ANTE: Abstracts in New Tech & Engineer 1966-2006/Aug
(c) 2006 CSA.

File 65:Inside Conferences 1993-2006/Sep 13
(c) 2006 BLDSC all rts. reserv.

File 94:JICST-EPlus 1985-2006/Jun W1
(c) 2006 Japan Science and Tech Corp(JST)

File 95:TEME-Technology & Management 1989-2006/Sep W2
(c) 2006 FIZ TECHNIK

File 99:Wilson Appl. Sci & Tech Abs 1983-2006/Jul
(c) 2006 The HW Wilson Co.

File 111:TGG Natl.Newspaper Index(SM) 1979-2006/Aug 31
(c) 2006 The Gale Group

File 144:Pascal 1973-2006/Aug W3
(c) 2006 INIST/CNRS

File 256:TecInfoSource 82-2006/Dec
(c) 2006 Info.Sources Inc

File 434:SciSearch(R) Cited Ref Sci 1974-1989/Dec
(c) 2006 The Thomson Corp

Your SELECT statement is:

s garbage(3n)collect??? and sarsa and (jvm or (java()virtual()machine))

Items	File
-----	-----
Examined 50 files	
Examined 100 files	
Examined 150 files	
Examined 200 files	
1	340: CLAIMS(R)/US Patent_1950-06/Sep 12
1	349: PCT FULLTEXT_1979-2006/UB=20060907UT=20060831
1	351: Derwent WPI_1963-2006/UD=200657
Examined 250 files	
Examined 300 files	
Examined 350 files	
Examined 400 files	
1	654: US Pat.Full._1976-2006/Sep 12
Examined 450 files	
Examined 500 files	
Examined 550 files	

4 files have one or more items; file list includes 566 files.

Set	Items	Description
S1	4	GARBAGE(3N)COLLECT??? AND SARSA AND (JVM OR (JAVA())VIRTUAL- ()MACHINE))

? show files

File 340:CLAIMS(R)/US Patent 1950-06/Sep 12

(c) 2006 IFI/CLAIMS(R)

File 349:PCT FULLTEXT 1979-2006/UB=20060907UT=20060831

(c) 2006 WIPO/Thomson

File 351:Derwent WPI 1963-2006/UD=200657

(c) 2006 The Thomson Corporation

File 654:US Pat.Full. 1976-2006/Sep 12

(c) Format only 2006 Dialog

Research
DatabasesBasic
SearchAdvanced
SearchVisual
Search[New Search](#) | [Folder](#) | [Preferences](#) | [Help](#)
[Choose Databases](#) | [Return to the USPTO NPL Page](#)[Sign In to My EBSCOhost](#)

Choose Databases

To search within a single database, click the database name listed below. To select more than one database to search, check the boxes next to the databases and click *Continue*.

Folder is empty.[Continue](#)**Business Source Corporate**

The world's largest, English-language, full text, periodical, business database provides corporations with nearly 2,800 quality magazines and journals, offering invaluable information on virtually every aspect of business. Areas covered include human resources, sales & marketing, finance, public relations, executive management, information management, manufacturing, and more. This database is updated on a daily basis.

[Title List](#)[More Information](#)[Enhanced Business Searching Interface](#)**Regional Business News**

This database provides comprehensive full text coverage for regional business publications. Regional Business News incorporates coverage of 75 business journals, newspapers and newswires from all metropolitan and rural areas within the United States. This database is updated on a daily basis.

[Title List](#)[More Information](#)**McClatchy-Tribune Collection**

The *McClatchy-Tribune Collection* includes a 90-day archive of approximately 290 newspapers from the McClatchy-Tribune Information Services. This database is updated daily.

[Title List](#)[More Information](#)**Academic Search Premier**

The world's largest academic multi-disciplinary database, *Academic Search Premier* provides full text for nearly 4,650 serials, including full text for more than 3,600 peer-reviewed titles. PDF backfiles to 1975 or further are available for well over one hundred journals, and searchable cited references are provided for more than 1,000 titles. This database is updated on a daily basis via EBSCOhost.

[Title List](#)[More Information](#)**Computer Science Index**

Computer Science Index offers abstracting and indexing of academic journals, professional publications, and other reference sources at the highest scholarly and technical levels of computer science. The collection covers more than 6,500 periodicals and books, with coverage going back to the mid 1960s.

[More Information](#)**Computer Source**

Computer Source provides researchers with the latest information and current trends in high technology. This database offers full text for nearly 300 publications and indexing and abstracts for nearly 450 publications.

[Title List](#)[More Information](#)**Internet and Personal Computing Abstracts**

Internet and Personal Computing Abstracts (formerly *Microcomputer Abstracts*) provides abstracts and indexing for literature related to personal computing products and developments in business, the Internet, the home, and all other applied areas. Over 400 of the most important trade publications, mainstream computer magazines, and professional journals are covered, including those that focus on specific topics such as Macintosh and Windows platforms, programming, web development and more.

[Title List](#) [More Information](#)☒ **Information Science & Technology Abstracts (ISTA)**

Information Science & Technology Abstracts is a leading database in the areas of information science. ISTA indexes journal articles from more than 450 publications plus books, research reports, and conference proceedings and patents, with ongoing comprehensive coverage of the most important periodicals in this field. Coverage dates back to the mid 1960s.

[More Information](#)[Continue](#)[Top of Page](#)[EBSCO Support Site](#)[Privacy Policy](#) | [Terms of Use](#) | [Copyright](#)

© 2006 EBSCO Industries, Inc. All rights reserved.

...JavaTM Virtual Machine (**JVM**) constructed by Appeal...different needs. Thus, a **garbage collection** technique and a **garbage collection** strategy that works well...different applications, various **garbage collection** techniques with different...

[<http://www.nada.kth.se/~hoffmann/jvm02.pdf>]

[similar results](#)

☐ 4. [01.covers \[PDF-13K\]](#)

Oct 2002

...learning contributes to **JVM** autonomous decision-making to per- form **garbage collection**. Given the ele- ments...State, new Action (**SARSA**) Eva explained how...Numerous aspects of **JVM** system architecture...manner. A **LIGHTWEIGHT JAVA VIRTUAL MACHINE** FOR A STACK-BASED MICROPROCESSOR...implementation of a **JVM**. The Ignite microprocessor...

[<http://www.usenix.org/events/jvm02/javaVM2002.pdf>]

[similar results](#)

☐ 5. [Proceedings of the \[PDF-59K\]](#)

Jul 2002

...**collection** in a modern **JVM**. Unlike some other **garbage**...with some phases of the **garbage collection**. For further reading about **garbage collection**, see references [2, 6...it comes to concurrent **garbage collection** in a **JVM**, is to decide...

[http://www.usenix.org/events/jvm02/full_papers/andreas...]

[similar results](#)

fast

[Downloads](#) | [Subscribe to News Updates](#) | [User Feedback](#) | [Advertising](#)
[Tell A Friend](#) | [Terms Of Service](#) | [Privacy Policy](#) | [Legal](#)

Powered by [FAST](#) © Elsevier 2006

[Sign in](#)

Google

[Web](#) [Images](#) [Video](#)^{New!} [News](#) [Maps](#) [more »](#)

sarsa AND (jvm OR "java virtual machine") AND

[Advanced Search](#)
[Preferences](#)The "AND" operator is unnecessary -- we include all search terms by default. [\[details\]](#)**Web** Results 1 - 7 of about 9 for **sarsa AND (jvm OR "java virtual machine") AND "garbage collection"**. (0.3

Tip: Try removing quotes from your search to get more results.

Sponsored Links

[PDF] [inside:](#)File Format: PDF/Adobe Acrobat - [View as HTML](#)Innovations in **JVM** memory allocation, and **garbage collection** were ... (**SARSA**) – Eva explained how after an. iterative and systematic process of deci- ...www.usenix.org/event/jvm02/JavaVM2002.pdf - [Similar pages](#)**Java jvm**Official Sun Site for Free J2EE SDK
World's Most Popular. Download Now.
java.sun.com**JVM '02 paper**The problem to solve is: how to design an automatic and learning decision process for more dynamic **garbage collection** in a modern **JVM**. ...www.usenix.org/event/jvm02/full_papers/andreasson/andreasson_html/index.html - 60k -[Cached](#) - [Similar pages](#)[\[More results from www.usenix.org \]](#)**[PDF]** **BEA WebLogic JRockit Adaptive Garbage Collection**File Format: PDF/Adobe Acrobat - [View as HTML](#)JRockit **JVM**. • See how interdisciplinary techniques open. doors to future development ...**SARSA***. • Observe the State. • Find and perform the best Action ...www.bea.com/.../Deployment_Management_Administration/1165_JRockit_Adaptive_Garbage_Collection.pdf - [Similar pages](#)**Java Virtual Machine 2002 Symposium, San Francisco, CA**Different **Garbage Collection** strategies were compared, ... of a **JVM** on a paper titled "A Lightweight **Java Virtual Machine** for a Stack-based Microprocessor" ...home.comcast.net/~osorio/jvm02Rev.html - 33k - [Cached](#) - [Similar pages](#)**[PDF]** **A Hybrid Reinforcement Learning Approach to Autonomic Resource ...**File Format: PDF/Adobe Acrobat - [View as HTML](#)of the specific learning algorithm that we use (**Sarsa**(0)), and ... sampling and **garbage collection** processes in Java. To dampen ...www.cs.utexas.edu/~nkj/papers/icac06.pdf - [Similar pages](#)**[PDF]** **Online Performance Management Using Hybrid Reinforcement Learning**File Format: PDF/Adobe Acrobat - [View as HTML](#)Line 5 derives from the well-known **Sarsa** learning. rule: ΔQ_s ... tribution (apart from **JVM garbage collection**). Given an estimated ...research.microsoft.com/workshops/sysml/papers/sysml-Tesauro.pdf - [Similar pages](#)**こんな毎日・・・ Recent SourceForge News**JamVM is a new **Java Virtual Machine** conforming to the **JVM** specification edition 2(blue ... It includes learning algorithms (TD, **Sarsa**, Q) as well as the ...machine.homeunix.org/weblog/archives/sf_news_2005-01-26.html - [Similar pages](#)*In order to show you the most relevant results, we have omitted some entries very similar to the 7 already displayed.**If you like, you can repeat the search with the omitted results included.*

Give feedback on RSS feeds for document recommendations in CiteSeer.



Find:

Documents

Citations

Searching for **PHRASE** garbage collection sarsa jvm.

Restrict to: Header Title Order by: Expected citations Hubs Usage Date Try: Google (CiteSeer) Google (Web) Yahoo! MSN CSB DBLP

No documents match Boolean query. Trying non-Boolean relevance query.

500 documents found. Order: relevance to query.

A Distributed Garbage Collector with Diffusion Tree.. - Moreau (1997) (Correct) (4 citations)

A Distributed **Garbage** Collector with Diffusion Tree Reorganisation and Abstract We present a new distributed **garbage collection** algorithm that is able to reorganise diffusion
www.ecs.soton.ac.uk/~lavm/papers/icfp98.ps

Linearity and Laziness - Wakeling, Runciman (1990) (Correct) (19 citations)

made by using linear types to reduce the amount of **garbage collection**. We also consider how the language linear types to reduce the amount of **garbage collection**. We also consider how the language and the
www.cs.york.ac.uk/~colin/papers/fpca91.ps.gz

Garbage Collecting the World - Lang, Queinnec, Piquer (1992) (Correct) (73 citations)

Garbage Collecting the World Bernard Lang
GC of all groups to which it belongs. **Garbage collection** on small groups reclaims quickly locally
ftp.inria.fr/INRIA/Projects/Atoll/Bernard.Lang/gc.ps.gz

Complementary Garbage Collector - Shogo Matsui (1995) (Correct) (2 citations)

Complementary **Garbage** Collector Shogo Matsui 1 Yoshio Tanaka 2
of both types: great efficiency of the **garbage collection** and ease of consistent implementation. This
ftp.ml.info.kanagawa-u.ac.jp/pub/matsui/ComplementaryGC.ps.gz

Garbage Collection in LOGFLOW - Norbert Podhorszki (Correct)

Garbage Collection in LOGFLOW Norbert Podhorszki
Garbage Collection in LOGFLOW Norbert Podhorszki Peter Kacsuk
ftp.lpsd.sztaki.hu/pub/lpsd/publications/LOGFLOW/gc_rep2.ps.gz

Computing the Subset Partial Order: Progress and Open Problems.. - Pritchard (Correct)

4111 P.Pritchard@cit.gu.edu.au Abstract A given **collection** of sets has a natural partial order induced by
ftp.cs.su.oz.au/symvonis/AWOCA-submissions/pritchard.ps

Sequential-System Factorization - Rath (Correct)

includes a processor, a heap with a stop-and-copy **garbage** collector and an allocator, and a dynamic RAM
its interface specification. Since the **garbage collection** procedure is embedded within the allocator
ftp.cs.indiana.edu/pub/techreports/TR457.ps.Z

Patrick C Hew - Phew Maths (Correct)

segmentation of the LCM points into digit sets or **garbage**, using the dimensions of the bounding box. 2
inspection of the box images. The original data **collection** gave no guidelines about how people were to
maths.uwa.edu.au/~phew/postgrad/diaries/initsegsd1str.ps.Z

The Berkeley Restaurant Project - Jurafsky, Wooters, Tajchman, Segal.. (1994) (Correct) (11 citations)

speech. The idea is to add a new word model, the **garbage** word, which is designed to match unknown words
The One With The Higher Probability. 8 Data **Collection** The Berp System Was Bootstrapped With A Wizard
ftp.icsi.berkeley.edu/pub/speech/papers/icslp94-berp.ps.Z

A More Efficient RMI for Java - Nester, Philippsen, Haumacher (1999) (Correct) (42 citations)

e.g. for special purpose communication hardware or **garbage** collectors. Currently available are optimized
protocols. As a by-product, a benchmark **collection** for RMI is presented. This **collection** -asked
Grande Forum from its first meeting -can guide **JVM** vendors in their performance optimizations. On PCs

Wed, 13 Sep 2006, 1:50:45 PM EST

Edit an existing query or
compose a new query in the
Search Query Display.

Search Query Display

NEW SEARCH

NEXT

Select a search number (#)
to:

- Add a query to the Search Query Display
- Combine search queries using AND, OR, or NOT
- Delete a search
- Run a search

Recent Search Queries

- #1 (((memory<in>metadata) <and>
(management<in>metadata))<and> (garbage<in>metadata))
<and> (pyr >= 1950 <and> pyr <= 2001)
- #2 (((memory<in>metadata) <and>
(management<in>metadata))<and> (sarsa<in>metadata))
<and> (pyr >= 1950 <and> pyr <= 2001)
- #3 (((memory<in>metadata) <and>
(management<in>metadata))<and>
(algorithms<in>metadata)) <and> (pyr >= 1950 <and> pyr <= 2001)
- #4 (((garbage<in>metadata) <and> (collection<in>metadata))
<and> (algorithms<in>metadata)) <and> (pyr >= 1950 <and> pyr <= 2001)
- #5 (((garbage<in>metadata) <and> (collection<in>metadata))
<and> (algorithms<in>metadata)) <and> (pyr >= 1950 <and> pyr <= 2001)
- #6 (((garbage<in>metadata) <and> (collection<in>metadata))
<and> (sarsa<in>metadata)) <and> (pyr >= 1950 <and> pyr <= 2001)
- #7 (((garbage<in>metadata) <and> (algorithm<in>metadata))
<and> (sarsa<in>metadata)) <and> (pyr >= 1950 <and> pyr <= 2001)
- #8 (((reinforcement<in>metadata) <and>
(algorithm<in>metadata))<and> (sarsa<in>metadata))
<and> (pyr >= 1950 <and> pyr <= 2001)
- #9 (((reinforcement<in>metadata) <and>
(algorithm<in>metadata))<and> (sarsa<in>metadata))
<and> (pyr >= 1950 <and> pyr <= 2001)
- #10 (((reinforcement<in>metadata) <and>
(garbage<in>metadata))<and> (virtual<in>metadata)) <and>
(pyr >= 1950 <and> pyr <= 2001)
- #11 (((java<in>metadata) <and> (garbage<in>metadata))<and>
(virtual<in>metadata)) <and> (pyr >= 1950 <and> pyr <= 2001)

(((calculating <in>metadata) <and> (memory<in>metadata))

#12 <and> (space<in>metadata) <and> (pyr >= 1950 <and> pyr
<= 2001)

#13 (((reward<in>metadata) <and> (memory<in>metadata))
<and> (space<in>metadata)) <and> (pyr >= 1950 <and> pyr
<= 2001)

Downloaded from

Indexed by
 Inspec

[Help](#) [Contact Us](#) [Privacy &](#)

© Copyright 2006 IEEE -



Terms used **performing memory space garbage collection**

Found **95,460** of **185,178**

Sort results by



[Save results to a Binder](#)

Try an [Advanced Search](#)

Try this search in [The ACM Guide](#)

Display results



[Search Tips](#)

☐ Open results in a new window

Results 1 - 20 of 200

Result page: [1](#) [2](#) [3](#) [4](#) [5](#) [6](#) [7](#) [8](#) [9](#) [10](#) [next](#)

Best 200 shown

Relevance scale ☐ ☐ ☐ ☐ ☐

1 [MC²: high-performance garbage collection for memory-constrained environments](#)



Narendran Sachindran, J. Eliot B. Moss, Emery D. Berger

October 2004 **ACM SIGPLAN Notices , Proceedings of the 19th annual ACM SIGPLAN conference on Object-oriented programming, systems, languages, and applications OOPSLA '04**, Volume 39 Issue 10

Publisher: ACM Press

Full text available: pdf(503.53 KB)

Additional Information: [full citation](#), [abstract](#), [references](#), [citations](#), [index terms](#)

Java is becoming an important platform for memory-constrained consumer devices such as PDAs and cellular phones, because it provides safety and portability. Since Java uses garbage collection, efficient garbage collectors that run in constrained memory are essential. Typical collection techniques used on these devices are mark-sweep and mark-compact. Mark-sweep collectors can provide good throughput and pause times but suffer from fragmentation. Mark-compact collectors prevent fragmentation, ...

Keywords: copying collector, generational collector, java, mark-compact, mark-copy, mark-sweep, memory-constrained copying

2 [New garbage collection algorithms and strategies: Automatic heap sizing: taking real memory into account](#)



Ting Yang, Matthew Hertz, Emery D. Berger, Scott F. Kaplan, J. Eliot B. Moss

October 2004 **Proceedings of the 4th international symposium on Memory management**

Publisher: ACM Press

Full text available: pdf(879.86 KB)

Additional Information: [full citation](#), [abstract](#), [references](#), [index terms](#)

Heap size has a huge impact on the performance of garbage collected applications. A heap that barely meets the application's needs causes excessive GC overhead, while a heap that exceeds physical memory induces paging. Choosing the best heap size *a priori* is impossible in multiprogrammed environments, where physical memory allocations to processes change constantly. We present an automatic heap-sizing algorithm applicable to different garbage collectors with only modest changes ...

Keywords: garbage collection, paging, virtual memory

3 [Tuning garbage collection for reducing memory system energy in an embedded java](#)



environment

G. Chen, R. Shetty, M. Kandemir, N. Vijaykrishnan, M. J. Irwin, M. Wolczko
November 2002 **ACM Transactions on Embedded Computing Systems (TECS)**, Volume 1
Issue 1

Publisher: ACM Press

Full text available: pdf(740.23 KB) Additional Information: [full citation](#), [abstract](#), [references](#), [citations](#), [index terms](#)

Java has been widely adopted as one of the software platforms for the seamless integration of diverse computing devices. Over the last year, there has been great momentum in adopting Java technology in devices such as cellphones, PDAs, and pagers where optimizing energy consumption is critical. Since, traditionally, the Java virtual machine (JVM), the cornerstone of Java technology, is tuned for performance, taking into account energy consumption requires reevaluation, and possibly redesign of t ...

Keywords: Garbage collector, Java Virtual Machine (JVM), K Virtual Machine (KVM), low power computing

4 Quantifying the performance of garbage collection vs. explicit memory management



Matthew Hertz, Emery D. Berger

October 2005 **ACM SIGPLAN Notices , Proceedings of the 20th annual ACM SIGPLAN conference on Object oriented programming, systems, languages, and applications OOPSLA '05**, Volume 40 Issue 10

Publisher: ACM Press

Full text available: pdf(1.51 MB) Additional Information: [full citation](#), [abstract](#), [references](#), [citations](#), [index terms](#)

Garbage collection yields numerous software engineering benefits, but its quantitative impact on performance remains elusive. One can compare the cost of *conservative* garbage collection to explicit memory management in C/C++ programs by linking in an appropriate collector. This kind of direct comparison is not possible for languages designed for garbage collection (e.g., Java), because programs in these languages naturally do not contain calls to free. Thus, the actual gap between the tim ...

Keywords: explicit memory management, garbage collection, oracular memory management, paging, performance analysis, throughput, time-space tradeoff

5 Cost-effective object space management for hardware-assisted real-time garbage collection



Kelvin D. Nilsen, William J. Schmidt

December 1992 **ACM Letters on Programming Languages and Systems (LOPLAS)**,
Volume 1 Issue 4


Publisher: ACM Press

Full text available: pdf(1.29 MB) Additional Information: [full citation](#), [abstract](#), [references](#), [citations](#), [index terms](#)

Modern object-oriented languages and programming paradigms require finer-grain division of memory than is provided by traditional paging and segmentation systems. This paper describes the design of an OSM (Object Space Manager) that allows partitioning of real memory on object, rather than page, boundaries. The time required by the OSM to create an object, or to find the beginning of an object given a pointer to any location within it, is approximately one memory cycle. Object sizes are lim ...

Keywords: automatic garbage collection, dynamic storage management, high-level language architectures, memory technologies, real-time and embedded systems, run-time environments

6 New garbage collection algorithms and strategies: Dynamic selection of application-specific garbage collectors

 Sunil Soman, Chandra Krintz, David F. Bacon

October 2004 **Proceedings of the 4th international symposium on Memory management**

Publisher: ACM Press

Full text available:  [pdf\(185.74 KB\)](#) Additional Information: [full citation](#), [abstract](#), [references](#), [index terms](#)

Much prior work has shown that the performance enabled by garbage collection (GC) systems is highly dependent upon the behavior of the application as well as on the available resources. That is, no single GC enables the best performance for all programs and all heap sizes. To address this limitation, we present the design, implementation, and empirical evaluation of a novel Java Virtual Machine (JVM) extension that facilitates dynamic switching between a number of very different and popular g ...

Keywords: Java, annotation, application-specific collection, dynamic selection, hot-swapping, virtual machine

7 Connectivity-based garbage collection

 Martin Hirzel, Amer Diwan, Matthew Hertz

October 2003 **ACM SIGPLAN Notices , Proceedings of the 18th annual ACM SIGPLAN conference on Object-oriented programing, systems, languages, and applications OOPSLA '03**, Volume 38 Issue 11


Publisher: ACM Press

Full text available:  [pdf\(521.65 KB\)](#) Additional Information: [full citation](#), [abstract](#), [references](#), [citings](#), [index terms](#)

We introduce a new family of connectivity-based garbage collectors (Cbgc) that are based on potential object-connectivity properties. The key feature of these collectors is that the placement of objects into partitions is determined by performing one of several forms of connectivity analyses on the program. This enables partial garbage collections, as in generational collectors, but without the need for any write barrier. The contributions of this paper are 1) a novel family of garbage c ...


Keywords: connectivity based garbage collection

8 Utlterior reference counting: fast garbage collection without a long wait

 Stephen M. Blackburn, Kathryn S. McKinley

October 2003 **ACM SIGPLAN Notices , Proceedings of the 18th annual ACM SIGPLAN conference on Object-oriented programing, systems, languages, and applications OOPSLA '03**, Volume 38 Issue 11

Publisher: ACM Press


Full text available:  [pdf\(218.61 KB\)](#) Additional Information: [full citation](#), [abstract](#), [references](#), [citings](#), [index terms](#)

General purpose garbage collectors have yet to combine short pause times with high throughput. For example, generational collectors can achieve high throughput. They have modest average pause times, but occasionally collect the whole heap and consequently incur long pauses. At the other extreme, concurrent collectors, including reference counting, attain short pause times but with significant performance penalties. This paper introduces a new hybrid collector that combines copying generational c ...


Keywords: Java, copying, generational hybrid, reference counting, ulterior reference

counting

9 Myths and realities: the performance impact of garbage collection

 Stephen M. Blackburn, Perry Cheng, Kathryn S. McKinley
June 2004 **ACM SIGMETRICS Performance Evaluation Review , Proceedings of the joint international conference on Measurement and modeling of computer systems SIGMETRICS '04/Performance '04**, Volume 32 Issue 1

Publisher: ACM Press

Full text available:  [pdf\(305.06 KB\)](#) Additional Information: [full citation](#), [abstract](#), [references](#), [citations](#), [index terms](#), [review](#)

This paper explores and quantifies garbage collection behavior for three whole heap collectors and generational counterparts: *copying semi-space*, *mark-sweep*, and *reference counting*, the canonical algorithms from which essentially all other collection algorithms are derived. Efficient implementations in MMTk, a Java memory management toolkit, in IBM's Jikes RVM share all common mechanisms to provide a clean experimental platform. Instrumentation separates collector and program behavior ...

Keywords: generational, java, mark-sweep, reference counting, semi-space

10 Garbage collection for a client-server persistent object store

 Laurent Amsaleg, Michael J. Franklin, Olivier Gruber
August 1999 **ACM Transactions on Computer Systems (TOCS)**, Volume 17 Issue 3


Publisher: ACM Press

Full text available:  [pdf\(267.18 KB\)](#) Additional Information: [full citation](#), [abstract](#), [references](#), [citations](#), [index terms](#), [review](#)

We describe an efficient server-based algorithm for garbage collecting persistent object stores in a client-server environment. The algorithm is incremental and runs concurrently with client transactions. Unlike previous algorithms, it does not hold any transactional locks on data and does not require callbacks to clients. It is fault-tolerant, but performs very little logging. The algorithm has been designed to be integrated into existing systems, and therefore it works with standard i ...

Keywords: client-server system, logging, persistent object-store, recovery

11 A unified theory of garbage collection

 David F. Bacon, Perry Cheng, V. T. Rajan
October 2004 **ACM SIGPLAN Notices , Proceedings of the 19th annual ACM SIGPLAN conference on Object-oriented programming, systems, languages, and applications OOPSLA '04**, Volume 39 Issue 10

Publisher: ACM Press

Full text available:  [pdf\(223.52 KB\)](#) Additional Information: [full citation](#), [abstract](#), [references](#), [index terms](#)

Tracing and reference counting are uniformly viewed as being fundamentally different approaches to garbage collection that possess very distinct performance properties. We have implemented high-performance collectors of both types, and in the process observed that the more we optimized them, the more similarly they behaved - that they seem to share some deep structure.

We present a formulation of the two algorithms that shows that they are in fact duals of each other. Intuitively, the ...

Keywords: graph algorithms, mark-and-sweep, reference counting, tracing

12 Garbage Collection of Linked Data Structures



Jacques Cohen

September 1981 **ACM Computing Surveys (CSUR)**, Volume 13 Issue 3

Publisher: ACM Press

Full text available: [pdf\(2.32 MB\)](#) Additional Information: [full citation](#), [references](#), [citations](#), [index terms](#)

13 Concurrent compacting garbage collection of a persistent heap



James O'Toole, Scott Nettles, David Gifford

December 1993 **ACM SIGOPS Operating Systems Review , Proceedings of the fourteenth ACM symposium on Operating systems principles SOSP '93**, Volume 27 Issue 5

Publisher: ACM Press

Full text available: [pdf\(1.50 MB\)](#) Additional Information: [full citation](#), [abstract](#), [references](#), [citations](#), [index terms](#)

We describe a replicating garbage collector for a persistent heap. The garbage collector cooperates with a transaction manager to provide safe and efficient transactional storage management. Clients read and write the heap in primary memory and can commit or abort their write operations. When write operations are committed they are preserved in stable storage and survive system failures. Clients can freely access the heap during garbage collection because the collector concurrently builds a comp ...

14 Garbage collection: Reducing generational copy reserve overhead with fallback compaction



Phil McGachey, Antony L. Hosking

June 2006 **Proceedings of the 2006 international symposium on Memory management ISMM '06**

Publisher: ACM Press

Full text available: [pdf\(817.15 KB\)](#) Additional Information: [full citation](#), [abstract](#), [references](#), [index terms](#)

As programming languages with managed runtimes become increasingly popular, it is essential that virtual machines are implemented efficiently. The performance of the memory management subsystem can be a defining factor in the performance of the virtual machine as a whole. We present a technique by which garbage collector performance can be improved. We describe an algorithm that combines a standard generational copying collector with a mark and compact collector. We observe that, since most object ...

Keywords: copying collector, garbage collection, generational collector, java, mark and compact

15 Controlling fragmentation and space consumption in the metronome, a real-time garbage collector for Java



David F. Bacon, Perry Cheng, V. T. Rajan

June 2003 **ACM SIGPLAN Notices , Proceedings of the 2003 ACM SIGPLAN conference on Language, compiler, and tool for embedded systems LCTES '03**, Volume 38 Issue 7

Publisher: ACM Press

Full text available: [pdf\(354.15 KB\)](#) Additional Information: [full citation](#), [abstract](#), [references](#), [citations](#), [index terms](#)

Now that the use of garbage collection in languages like Java is becoming widely accepted

due to the safety and software engineering benefits it provides, there is significant interest in applying garbage collection to hard real-time systems. Past approaches have generally suffered from one of two major flaws: either they were not provably real-time, or they imposed large space overheads to meet the real-time bounds. Our previous work [3] presented the Metronome, a mostly non-copying real-time co ...

Keywords: compaction, cost model, fragmentation, space bounds

16 Error-free garbage collection traces: how to cheat and not get caught


 Matthew Hertz, Stephen M Blackburn, J Eliot B Moss, Kathryn S. McKinley, Darko Stefanović
June 2002 **ACM SIGMETRICS Performance Evaluation Review , Proceedings of the 2002 ACM SIGMETRICS international conference on Measurement and modeling of computer systems SIGMETRICS '02**, Volume 30 Issue 1

Publisher: ACM Press

Full text available:  pdf(105.06 KB) Additional Information: [full citation](#), [abstract](#), [references](#), [citations](#)

Programmers are writing a large and rapidly growing number of programs in object-oriented languages such as Java that require garbage collection (GC). To explore the design and evaluation of GC algorithms quickly, researchers are using simulation based on traces of object allocation and lifetime behavior. The *brute force* method generates perfect traces using a whole-heap GC at every potential GC point in the program. Because this process is prohibitively expensive, researchers often use < ...

17 Concurrent garbage collection using hardware-assisted profiling

 Timothy H. Heil, James E. Smith
October 2000 **ACM SIGPLAN Notices , Proceedings of the 2nd international symposium on Memory management ISMM '00**, Volume 36 Issue 1

Publisher: ACM Press

Full text available:  pdf(1.74 MB) Additional Information: [full citation](#), [abstract](#), [citations](#), [index terms](#)

In the presence of on-chip multithreading, a Virtual Machine (VM) implementation can readily take advantage of *service threads* for enhancing performance by performing tasks such as profile collection and analysis, dynamic optimization, and garbage collection concurrently with program execution. In this context, a hardware-assisted profiling mechanism is proposed. The *Relational Profiling Architecture* (RPA) is designed from the top down. RPA is based on a relational model similar ...

18 Beltway: getting around garbage collection gridlock

 Stephen M Blackburn, Richard Jones, Kathryn S. McKinley, J Eliot B Moss
May 2002 **ACM SIGPLAN Notices , Proceedings of the ACM SIGPLAN 2002 Conference on Programming language design and implementation PLDI '02**, Volume 37 Issue 5

Publisher: ACM Press

Full text available:  pdf(184.50 KB) Additional Information: [full citation](#), [abstract](#), [references](#), [citations](#), [index terms](#)

We present the design and implementation of a new garbage collection framework that significantly generalizes existing copying collectors. The *Beltway* framework exploits and separates object age and incrementality. It groups objects in one or more increments on queues called *belts*, collects belts independently, and collects increments on a belt in first-in-first-out order. We show that Beltway configurations, selected by command line options, act and perform the same as semi-space, ...

Keywords: Java, beltway, copying collection, generational collection

19 Garbage collection on multiprocessors: Task-aware garbage collection in a multi-tasking virtual machine



Sunil Soman, Laurent Daynès, Chandra Krintz

June 2006 **Proceedings of the 2006 international symposium on Memory management ISMM '06**

Publisher: ACM Press

Full text available: [pdf\(125.22 KB\)](#) Additional Information: [full citation](#), [abstract](#), [references](#), [index terms](#)

A multi-tasking virtual machine (MVM) executes multiple programs in isolation, within a single operating system process. The goal of a MVM is to improve startup time, overall system throughput, and performance, by effective reuse and sharing of system resources across programs (tasks). However, multitasking also mandates a memory management system capable of offering a guarantee of isolation with respect to garbage collection costs, accounting of memory usage, and timely reclamation of heap reso ...

Keywords: java, multi-tasking, resource reclamation, task-aware garbage collection, virtual machine

20 An effective garbage collection strategy for parallel programming languages on large scale distributed-memory machines



Kenjiro Taura, Akinori Yonezawa

June 1997 **ACM SIGPLAN Notices , Proceedings of the sixth ACM SIGPLAN symposium on Principles and practice of parallel programming PPOPP '97**, Volume 32 Issue 7

Publisher: ACM Press

Full text available: [pdf\(1.43 MB\)](#) Additional Information: [full citation](#), [abstract](#), [references](#), [citings](#), [index terms](#)

This paper describes the design and implementation of a garbage collection scheme on large-scale distributed-memory computers and reports various experimental results. The collector is based on the conservative GC library by Boehm & Weiser. Each processor traces local pointers using the GC library while traversing remote pointers by exchanging "mark messages" between processors. It exhibits a promising performance---in the most space-intensive settings we tested, the total collection ove ...

Results 1 - 20 of 200

Result page: [1](#) [2](#) [3](#) [4](#) [5](#) [6](#) [7](#) [8](#) [9](#) [10](#) [next](#)

The ACM Portal is published by the Association for Computing Machinery. Copyright © 2006 ACM, Inc.

[Terms of Usage](#) [Privacy Policy](#) [Code of Ethics](#) [Contact Us](#)

Useful downloads: [Adobe Acrobat](#) [QuickTime](#) [Windows Media Player](#) [Real Player](#)



Terms used

2001 system wide application specific memory space

Found 127,709 of 185,178

Sort results by

relevance

[Save results to a Binder](#)

[Try an Advanced Search](#)

[Try this search in The ACM Guide](#)

Display results

expanded form

[Search Tips](#)

☐ Open results in a new window

Results 1 - 20 of 200

Result page: [1](#) [2](#) [3](#) [4](#) [5](#) [6](#) [7](#) [8](#) [9](#) [10](#) [next](#)

Best 200 shown

Relevance scale ☐ ☐ ☐ ☐ ☐

1 [Computing curricula 2001](#)



September 2001 **Journal on Educational Resources in Computing (JERIC)**

Publisher: ACM Press

Full text available: [pdf\(613.63 KB\)](#)

[html\(2.78 KB\)](#)

Additional Information: [full citation](#), [references](#), [citations](#), [index terms](#)

2 [System support for pervasive applications](#)



Robert Grimm, Janet Davis, Eric Lemar, Adam Macbeth, Steven Swanson, Thomas Anderson, Brian Bershad, Gaetano Borriello, Steven Gribble, David Wetherall

November 2004 **ACM Transactions on Computer Systems (TOCS)**, Volume 22 Issue 4

Publisher: ACM Press

Full text available: [pdf\(1.82 MB\)](#)

Additional Information: [full citation](#), [abstract](#), [references](#), [index terms](#)

Pervasive computing provides an attractive vision for the future of computing. Computational power will be available everywhere. Mobile and stationary devices will dynamically connect and coordinate to seamlessly help people in accomplishing their tasks. For this vision to become a reality, developers must build applications that constantly adapt to a highly dynamic computing environment. To make the developers' task feasible, we present a system architecture for pervasive computing, called & ...

Keywords: Asynchronous events, checkpointing, discovery, logic/operation pattern, migration, one.world, pervasive computing, structured I/O, tuples, ubiquitous computing

3 [Data and memory optimization techniques for embedded systems](#)



P. R. Panda, F. Catthoor, N. D. Dutt, K. Danckaert, E. Brockmeyer, C. Kulkarni, A. Vandercappelle, P. G. Kjeldsberg

April 2001 **ACM Transactions on Design Automation of Electronic Systems (TODAES)**, Volume 6 Issue 2

Publisher: ACM Press

Full text available: [pdf\(339.91 KB\)](#)

Additional Information: [full citation](#), [abstract](#), [references](#), [citations](#), [index terms](#)

We present a survey of the state-of-the-art techniques used in performing data and memory-related optimizations in embedded systems. The optimizations are targeted directly or indirectly at the memory subsystem, and impact one or more out of three

important cost metrics: area, performance, and power dissipation of the resulting implementation. We first examine architecture-independent optimizations in the form of code transformations. We next cover a broad spectrum of optimizati ...

Keywords: DRAM, SRAM, address generation, allocation, architecture exploration, code transformation, data cache, data optimization, high-level synthesis, memory architecture customization, memory power dissipation, register file, size estimation, survey

4 Data remapping for design space optimization of embedded memory systems



Rodric M. Rabbah, Krishna V. Palem

May 2003 **ACM Transactions on Embedded Computing Systems (TECS)**, Volume 2 Issue 2

Publisher: ACM Press

Full text available: pdf(885.05 KB) Additional Information: [full citation](#), [abstract](#), [references](#), [citations](#), [index terms](#)

In this article, we present a novel linear time algorithm for *data remapping*, that is, (i) lightweight; (ii) fully automated; and (iii) applicable in the context of pointer-centric programming languages with dynamic memory allocation support. All previous work in this area lacks one or more of these features. We proceed to demonstrate a *novel application of this algorithm as a key step in optimizing the design of an embedded memory system*. Specifically, we show that by virtue of lo ...

Keywords: Design space exploration, caches, compiler optimization, data remapping, embedded systems, memory hierarchy, memory subsystem

5 Level set and PDE methods for computer graphics



David Breen, Ron Fedkiw, Ken Museth, Stanley Osher, Guillermo Sapiro, Ross Whitaker

August 2004 **Proceedings of the conference on SIGGRAPH 2004 course notes SIGGRAPH '04**

Publisher: ACM Press

Full text available: pdf(17.07 MB) Additional Information: [full citation](#), [abstract](#), [citations](#)

Level set methods, an important class of partial differential equation (PDE) methods, define dynamic surfaces implicitly as the level set (iso-surface) of a sampled, evolving nD function. The course begins with preparatory material that introduces the concept of using partial differential equations to solve problems in computer graphics, geometric modeling and computer vision. This will include the structure and behavior of several different types of differential equations, e.g. the level set eq ...

6 New garbage collection algorithms and strategies: Dynamic selection of application-specific garbage collectors



Sunil Soman, Chandra Krantz, David F. Bacon

October 2004 **Proceedings of the 4th international symposium on Memory management**

Publisher: ACM Press


Full text available: pdf(185.74 KB) Additional Information: [full citation](#), [abstract](#), [references](#), [index terms](#)

Much prior work has shown that the performance enabled by garbage collection (GC) systems is highly dependent upon the behavior of the application as well as on the available resources. That is, no single GC enables the best performance for all programs and all heap sizes. To address this limitation, we present the design, implementation, and empirical evaluation of a novel Java Virtual Machine (JVM) extension that facilitates dynamic switching between a number of very different and popular g ...

Keywords: Java, annotation, application-specific collection, dynamic selection, hot-

swapping, virtual machine

7 GPGPU: general purpose computation on graphics hardware

 David Luebke, Mark Harris, Jens Krüger, Tim Purcell, Naga Govindaraju, Ian Buck, Cliff Woolley, Aaron Lefohn


August 2004 **Proceedings of the conference on SIGGRAPH 2004 course notes**
SIGGRAPH '04

Publisher: ACM Press

Full text available:  [pdf\(63.03 MB\)](#) Additional Information: [full citation](#), [abstract](#)

The graphics processor (GPU) on today's commodity video cards has evolved into an extremely powerful and flexible processor. The latest graphics architectures provide tremendous memory bandwidth and computational horsepower, with fully programmable vertex and pixel processing units that support vector operations up to full IEEE floating point precision. High level languages have emerged for graphics hardware, making this computational power accessible. Architecturally, GPUs are highly parallel s ...

8 Real-time shading

 Marc Olano, Kurt Akeley, John C. Hart, Wolfgang Heidrich, Michael McCool, Jason L. Mitchell, Randi Rost

August 2004 **Proceedings of the conference on SIGGRAPH 2004 course notes**
SIGGRAPH '04

Publisher: ACM Press

Full text available:  [pdf\(7.39 MB\)](#) Additional Information: [full citation](#), [abstract](#)

Real-time procedural shading was once seen as a distant dream. When the first version of this course was offered four years ago, real-time shading was possible, but only with one-of-a-kind hardware or by combining the effects of tens to hundreds of rendering passes. Today, almost every new computer comes with graphics hardware capable of interactively executing shaders of thousands to tens of thousands of instructions. This course has been redesigned to address today's real-time shading capabili ...

9 Access pattern-based memory and connectivity architecture exploration

 Peter Grun, Nikil Dutt, Alex Nicolau

February 2003 **ACM Transactions on Embedded Computing Systems (TECS)**, Volume 2
Issue 1

Publisher: ACM Press

Full text available:  [pdf\(857.06 KB\)](#) Additional Information: [full citation](#), [abstract](#), [references](#), [index terms](#)

Memory accesses represent a major bottleneck in embedded systems power and performance. Traditionally, designers tried to alleviate this problem by relying on a simple cache hierarchy, or a limited use of special purpose memory modules such as stream buffers. Although real-life applications contain a large number of memory references to a diverse set of data structures, a significant percentage of all memory accesses in the application are generated from a few memory instructions that exhibit pr ...

Keywords: Memory, access patterns, architecture exploration

10 Prototyping time- and space-efficient computations of algebraic operations over dynamically reconfigurable systems modeled by rewriting-logic

 M. Ayala-Rincón, C. H. Llanos, R. P. Jacobi, R. W. Hartenstein

April 2006 **ACM Transactions on Design Automation of Electronic Systems (TODAES)**,
Volume 11 Issue 2

Publisher: ACM Press

Full text available:  [pdf\(838.46 KB\)](#) Additional Information: [full citation](#), [abstract](#), [references](#), [index terms](#)

Many algebraic operations can be efficiently implemented as pipe networks in arrays of functional units such as systolic arrays that provide a large amount of parallelism. However, the applicability of classical systolic arrays is restricted to problems with strictly regular data dependencies yielding only arrays with uniform linear pipes. This limitation can be circumvented by using reconfigurable systolic arrays or reconfigurable data path arrays, where the node interconnections and operations ...

Keywords: Fast Fourier Transform (FFT), Term Rewriting Systems (TRS), algebraic manipulation, dynamically reconfigurable systems, reconfigurable computing, rewriting-logic, systolic arrays


11 Rapid design space exploration of heterogeneous embedded systems using symbolic

search and multi-granular simulation

S. Mohanty, V. K. Prasanna, S. Neema, J. Davis

June 2002 **ACM SIGPLAN Notices , Proceedings of the joint conference on Languages, compilers and tools for embedded systems: software and compilers for embedded systems LCTES/SCOPES '02**, Volume 37 Issue 7

Publisher: ACM Press

Full text available:  pdf(356.42 KB) Additional Information: [full citation](#), [abstract](#), [references](#), [citations](#), [index terms](#)

In addition to integrating different Intellectual Property cores, heterogeneous embedded systems provide several architecture knobs such as voltage, operating frequency, configuration, etc. that can be varied to optimize performance. Such flexibilities results in a large design space making system optimization a very challenging task. Moreover, such systems operate in mobile and other power constrained environments. Therefore, in addition to rapid exploration of a large design space a designer h ...


Keywords: binary decision diagram, design space, model integrated computing, modeling, multi-granular simulation, performance estimation, symbolic search

12 Consistency and replication: Application specific data replication for edge services

Lei Gao, Mike Dahlin, Amol Nayate, Jiandan Zheng, Arun Iyengar

May 2003 **Proceedings of the 12th international conference on World Wide Web**

Publisher: ACM Press

Full text available:  pdf(476.22 KB) Additional Information: [full citation](#), [abstract](#), [references](#), [citations](#), [index terms](#)

The emerging edge services architecture promises to improve the availability and performance of web services by replicating servers at geographically distributed sites. A key challenge in such systems is data replication and consistency so that edge server code can manipulate shared data without incurring the availability and performance penalties that would be incurred by accessing a traditional centralized database. This paper explores using a distributed object architecture to build an edge s ...

Keywords: availability, data replication, distributed objects, edge services, performance, wide area networks (WAN)

13 The elements of nature: interactive and realistic techniques

Oliver Deussen, David S. Ebert, Ron Fedkiw, F. Kenton Musgrave, Przemyslaw Prusinkiewicz, Doug Roble, Jos Stam, Jerry Tessendorf

August 2004 **Proceedings of the conference on SIGGRAPH 2004 course notes SIGGRAPH '04**

Publisher: ACM Press

Full text available:  [pdf\(17.65 MB\)](#) Additional Information: [full citation](#), [abstract](#)

This updated course on simulating natural phenomena will cover the latest research and production techniques for simulating most of the elements of nature. The presenters will provide movie production, interactive simulation, and research perspectives on the difficult task of photorealistic modeling, rendering, and animation of natural phenomena. The course offers a nice balance of the latest interactive graphics hardware-based simulation techniques and the latest physics-based simulation techni ...

14 A survey of research and practices of Network-on-chip



Tobias Bjerregaard, Shankar Mahadevan

June 2006 **ACM Computing Surveys (CSUR)**, Volume 38 Issue 1

Publisher: ACM Press

Full text available:  [pdf\(1.41 MB\)](#) Additional Information: [full citation](#), [abstract](#), [references](#), [index terms](#)

The scaling of microchip technologies has enabled large scale systems-on-chip (SoC). Network-on-chip (NoC) research addresses global communication in SoC, involving (i) a move from computation-centric to communication-centric design and (ii) the implementation of scalable communication structures. This survey presents a perspective on existing NoC research. We define the following abstractions: system, network adapter, network, and link to explain and structure the fundamental concepts. First, r ...

Keywords: Chip-area networks, GALS, GSI design, NoC, OCP, SoC, ULSI design, communication abstractions, communication-centric design, interconnects, network-on-chip, on-chip communication, sockets, system-on-chip


15 LegionFS: a secure and scalable file system supporting cross-domain high-performance applications



Brian S. White, Michael Walker, Marty Humphrey, Andrew S. Grimshaw

November 2001 **Proceedings of the 2001 ACM/IEEE conference on Supercomputing (CDROM)**

Publisher: ACM Press

Full text available:  [pdf\(499.88 KB\)](#) Additional Information: [full citation](#), [abstract](#), [references](#), [citings](#), [index terms](#)

Realizing that current file systems can not cope with the diverse requirements of wide-area collaborations, researchers have developed data access facilities to meet their needs. Recent work has focused on comprehensive data access architectures. In order to fulfill the evolving requirements in this environment, we suggest a more fully-integrated architecture built upon the fundamental tenets of naming, security, scalability, extensibility, and adaptability. These form the underpinning of the Le ...

16 Memory management and address optimization in embedded systems: Exploiting shared scratch pad memory space in embedded multiprocessor systems



Mahmut Kandemir, J. Ramanujam, A. Choudhary

June 2002 **Proceedings of the 39th conference on Design automation**


Publisher: ACM Press

Full text available:  [pdf\(254.79 KB\)](#) Additional Information: [full citation](#), [abstract](#), [references](#), [citings](#), [index terms](#)

In this paper, we present a compiler strategy to optimize data accesses in regular array-intensive applications running on embedded multiprocessor environments. Specifically, we propose an optimization algorithm that targets the reduction of extra off-chip memory accesses caused by inter-processor communication. This is achieved by increasing the application-wide reuse of data that resides in the scratch-pad memories of processors. Our experimental results obtained on four array-intensive image ...

Keywords: access patterns, compiler optimizations, data tiles, embedded multiprocessors, energy consumption, memories, scratch pad

17 System-wide compaction and specialization of the linux kernel

 Dominique Chagnet, Bjorn De Sutter, Bruno De Bus, Ludo Van Put, Koen De Bosschere
June 2005 **ACM SIGPLAN Notices , Proceedings of the 2005 ACM SIGPLAN/SIGBED conference on Languages, compilers, and tools for embedded systems**
LCTES '05, Volume 40 Issue 7

Publisher: ACM Press

Full text available:  [pdf\(895.17 KB\)](#) Additional Information: [full citation](#), [abstract](#), [references](#), [citations](#), [index terms](#)

The limited built-in configurability of Linux can lead to expensive code size overhead when it is used in the embedded market. To overcome this problem, we propose the application of link-time compaction and specialization techniques that exploit the *a priori* known, fixed run-time environment of many embedded systems. In experimental setups based on the ARM XScale and i386 platforms, the proposed techniques are able to reduce the kernel memory footprint with over 16%. We also show how rel ...

Keywords: compaction, linux kernel, operating system, specialization, system calls

18 Power reduction techniques for microprocessor systems

 Vasanth Venkatachalam, Michael Franz
September 2005 **ACM Computing Surveys (CSUR)**, Volume 37 Issue 3

Publisher: ACM Press

Full text available:  [pdf\(602.33 KB\)](#) Additional Information: [full citation](#), [abstract](#), [references](#), [index terms](#)

Power consumption is a major factor that limits the performance of computers. We survey the "state of the art" in techniques that reduce the total power consumed by a microprocessor system over time. These techniques are applied at various levels ranging from circuits to architectures, architectures to system software, and system software to applications. They also include holistic approaches that will become more important over the next decade. We conclude that power management is a ...

Keywords: Energy dissipation, power reduction

19 External memory algorithms and data structures: dealing with massive data

 Jeffrey Scott Vitter
June 2001 **ACM Computing Surveys (CSUR)**, Volume 33 Issue 2

Publisher: ACM Press

Full text available:  [pdf\(828.46 KB\)](#) Additional Information: [full citation](#), [abstract](#), [references](#), [citations](#), [index terms](#)

Data sets in large applications are often too massive to fit completely inside the computers internal memory. The resulting input/output communication (or I/O) between fast internal memory and slower external memory (such as disks) can be a major performance bottleneck. In this article we survey the state of the art in the design and analysis of external memory (or EM) algorithms and data structures, where the goal is to exploit locality in order to reduce the I/O costs. We consider a varie ...

Keywords: B-tree, I/O, batched, block, disk, dynamic, extendible hashing, external memory, hierarchical memory, multidimensional access methods, multilevel memory, online, out-of-core, secondary storage, sorting

20 Fast and flexible application-level networking on exokernel systems

 Gregory R. Ganger, Dawson R. Engler, M. Frans Kaashoek, Hector M. Briceño, Russell Hunt, Thomas Pinckney

February 2002 **ACM Transactions on Computer Systems (TOCS)**, Volume 20 Issue 1

Publisher: ACM Press

Full text available:  [pdf\(500.67 KB\)](#) Additional Information: [full citation](#), [abstract](#), [references](#), [citings](#), [index terms](#)

Application-level networking is a promising software organization for improving performance and functionality for important network services. The Xok/ExOS exokernel system includes application-level support for standard network services, while at the same time allowing application writers to specialize networking services. This paper describes how Xok/ExOS's kernel mechanisms and library operating system organization achieve this flexibility, and retrospectively shares our experiences an ...

Keywords: Extensible systems, OS structure, fast servers, network services

Results 1 - 20 of 200

Result page: [1](#) [2](#) [3](#) [4](#) [5](#) [6](#) [7](#) [8](#) [9](#) [10](#) [next](#)

The ACM Portal is published by the Association for Computing Machinery. Copyright © 2006 ACM, Inc..

[Terms of Usage](#) [Privacy Policy](#) [Code of Ethics](#) [Contact Us](#)

Useful downloads:  [Adobe Acrobat](#)  [QuickTime](#)  [Windows Media Player](#)  [Real Player](#)



USPTO

[Subscribe \(Full Service\)](#) [Register \(Limited Service, Free\)](#) [Login](#)

Search: ☒ The ACM Digital Library ☐ The Guide

sarsa garbage collection

SEARCH

THE ACM DIGITAL LIBRARY



[Feedback](#) [Report a problem](#) [Satisfaction survey](#)

Terms used **sarsa garbage collection**

Found **4,710** of **185,178**

Sort results
by

relevance ☒

Display
results

expanded form ☒



[Save results to a Binder](#)



[Search Tips](#)

☐ Open results in a new
window

Try an [Advanced Search](#)

Try this search in [The ACM Guide](#)

Results 1 - 20 of 200

Result page: [1](#) [2](#) [3](#) [4](#) [5](#) [6](#) [7](#) [8](#) [9](#) [10](#) [next](#)

Best 200 shown

Relevance scale ☐ ☐ ☐ ☐ ☐

1 [Controlling garbage collection and heap growth to reduce the execution time of Java applications](#)



Tim Brecht, Eshrat Arjomandi, Chang Li, Hang Pham

September 2006 **ACM Transactions on Programming Languages and Systems (TOPLAS)**, Volume 28 Issue 5

Publisher: ACM Press

Full text available: pdf(335.24 KB) Additional Information: [full citation](#), [abstract](#), [references](#), [index terms](#)

In systems that support garbage collection, a tension exists between collecting garbage too frequently and not collecting it frequently enough. Garbage collection that occurs too frequently may introduce unnecessary overheads at the risk of not collecting much garbage during each cycle. On the other hand, collecting garbage too infrequently can result in applications that execute with a large amount of virtual memory (i.e., with a large footprint) and suffer from increased execution times due to ...

Keywords: Garbage collection, Java, heap growth, implementation, memory management, performance measurement, programming languages

2 [Garbage collection: A true hardware read barrier](#)



Matthias Meyer

June 2006 **Proceedings of the 2006 international symposium on Memory management ISMM '06**

Publisher: ACM Press

Full text available: pdf(991.66 KB) Additional Information: [full citation](#), [abstract](#), [references](#), [index terms](#)

Read barriers synchronize compacting garbage collection and application processing in a simple yet elegant way. Unfortunately, read barrier checks are expensive to implement in software, and even with hardware support, the clustering of read barrier faults irregularly impairs application progress to an unacceptable extent. For this reason, read barriers are often considered unsuitable for hard real-time systems. In this paper, we introduce a novel hardware read barrier design for an object-based ...

Keywords: hardware support, object-based processor architecture, read barrier, real-time garbage collection

3

[Myths and realities: the performance impact of garbage collection](#)



Stephen M. Blackburn, Perry Cheng, Kathryn S. McKinley

June 2004 **ACM SIGMETRICS Performance Evaluation Review , Proceedings of the joint international conference on Measurement and modeling of computer systems SIGMETRICS '04/Performance '04**, Volume 32 Issue 1

Publisher: ACM Press

Full text available: pdf(305.06 KB)

Additional Information: [full citation](#), [abstract](#), [references](#), [citings](#), [index terms](#), [review](#)

This paper explores and quantifies garbage collection behavior for three whole heap collectors and generational counterparts: *copying semi-space*, *mark-sweep*, and *reference counting*, the canonical algorithms from which essentially all other collection algorithms are derived. Efficient implementations in MMTk, a Java memory management toolkit, in IBM's Jikes RVM share all common mechanisms to provide a clean experimental platform. Instrumentation separates collector and program behavior ...

Keywords: generational, java, mark-sweep, reference counting, semi-space

4 Quantifying the performance of garbage collection vs. explicit memory management



Matthew Hertz, Emery D. Berger

October 2005 **ACM SIGPLAN Notices , Proceedings of the 20th annual ACM SIGPLAN conference on Object oriented programming, systems, languages, and applications OOPSLA '05**, Volume 40 Issue 10

Publisher: ACM Press

Full text available: pdf(1.51 MB)

Additional Information: [full citation](#), [abstract](#), [references](#), [citings](#), [index terms](#)

Garbage collection yields numerous software engineering benefits, but its quantitative impact on performance remains elusive. One can compare the cost of *conservative* garbage collection to explicit memory management in C/C++ programs by linking in an appropriate collector. This kind of direct comparison is not possible for languages designed for garbage collection (e.g., Java), because programs in these languages naturally do not contain calls to free. Thus, the actual gap between the time ...

Keywords: explicit memory management, garbage collection, oracular memory management, paging, performance analysis, throughput, time-space tradeoff

5 Partition selection policies in object database garbage collection



Jonathan E. Cook, Alexander L. Wolf, Benjamin G. Zorn

May 1994 **ACM SIGMOD Record , Proceedings of the 1994 ACM SIGMOD international conference on Management of data SIGMOD '94**, Volume 23 Issue 2

Publisher: ACM Press

Full text available: pdf(1.23 MB)

Additional Information: [full citation](#), [abstract](#), [references](#), [citings](#), [index terms](#)

The automatic reclamation of storage for unreferenced objects is very important in object databases. Existing language system algorithms for automatic storage reclamation have been shown to be inappropriate. In this paper, we investigate methods to improve the performance of algorithms for automatic storage reclamation of object databases. These algorithms are based on a technique called partitioned garbage collection, in which a subset of the entire database ...

6 Robust, distributed references and acyclic garbage collection



Marc Shapiro, Peter Dickman, David Plainfossé

October 1992 **Proceedings of the eleventh annual ACM symposium on Principles of distributed computing PODC '92**

Publisher: ACM Press

Full text available:  pdf(1.27 MB)

Additional Information: [full citation](#), [references](#), [citations](#), [index terms](#), [review](#)

7 Atomic garbage collection: managing a stable heap



Elliot Kolodner, Barbara Liskov, William Weihl

June 1989 **ACM SIGMOD Record , Proceedings of the 1989 ACM SIGMOD international conference on Management of data SIGMOD '89**, Volume 18 Issue 2

Publisher: ACM Press

Full text available:  pdf(1.53 MB)

Additional Information: [full citation](#), [abstract](#), [references](#), [citations](#), [index terms](#)

Modern database systems use transactions to achieve a high degree of fault-tolerance. Many modern programming languages and systems provide garbage collected heap storage, which frees the programmer from the job of explicitly deallocating storage. In this paper we describe integrated garbage collection and recovery algorithms for managing a stable heap in which accessible objects survive both system crashes and media failures. A garbage collector typically both m ...

8 Garbage collection without paging



Matthew Hertz, Yi Feng, Emery D. Berger

June 2005 **ACM SIGPLAN Notices , Proceedings of the 2005 ACM SIGPLAN conference on Programming language design and implementation PLDI '05**, Volume 40 Issue 6

Publisher: ACM Press

Full text available:  pdf(231.14 KB)

Additional Information: [full citation](#), [abstract](#), [references](#), [citations](#), [index terms](#)

Garbage collection offers numerous software engineering advantages, but interacts poorly with virtual memory managers. Existing garbage collectors require far more pages than the application's working set and touch pages without regard to which ones are in memory, especially during full-heap garbage collection. The resulting paging can cause throughput to plummet and pause times to spike up to seconds or even minutes. We present a garbage collector that avoids paging. This *bookmarking collect* ...

Keywords: *bookmarking collection, garbage collection, generational collection, memory pressure, paging, virtual memory*

9 Connectivity-based garbage collection



Martin Hirzel, Amer Diwan, Matthew Hertz

October 2003 **ACM SIGPLAN Notices , Proceedings of the 18th annual ACM SIGPLAN conference on Object-oriented programing, systems, languages, and applications OOPSLA '03**, Volume 38 Issue 11

Publisher: ACM Press

Full text available:  pdf(521.65 KB)

Additional Information: [full citation](#), [abstract](#), [references](#), [citations](#), [index terms](#)

We introduce a new family of connectivity-based garbage collectors (Cbgc) that are based on potential object-connectivity properties. The key feature of these collectors is that the placement of objects into partitions is determined by performing one of several forms of connectivity analyses on the program. This enables partial garbage collections, as in generational collectors, but without the need for any write barrier. The contributions of this paper are 1) a novel family of garbage c ...

Keywords: connectivity based garbage collection

10 Mostly concurrent garbage collection revisited



Katherine Barabash, Yoav Ossia, Erez Petrank

October 2003 **ACM SIGPLAN Notices , Proceedings of the 18th annual ACM SIGPLAN conference on Object-oriented programming, systems, languages, and applications OOPSLA '03**, Volume 38 Issue 11

Publisher: ACM Press

Full text available: [pdf\(279.42 KB\)](#) Additional Information: [full citation](#), [abstract](#), [references](#), [citations](#), [index terms](#)

The *mostly concurrent garbage collection* was presented in the seminal paper of Boehm et al. With the deployment of Java as a portable, secure and concurrent programming language, the mostly concurrent garbage collector turned out to be an excellent solution for Java's garbage collection task. The use of this collector is reported for several modern production Java Virtual Machines and it has been investigated further in academia. In this paper, we present a modification of the mostly concu ...

Keywords: JVM, Java, concurrent garbage collection, garbage collection, incremental garbage collection

11 Incremental distribution of timestamp packets: a new approach to distributed garbage collection



M. Schelvis

September 1989 **ACM SIGPLAN Notices , Conference proceedings on Object-oriented programming systems, languages and applications OOPSLA '89**, Volume 24 Issue 10

Publisher: ACM Press

Full text available: [pdf\(1.20 MB\)](#) Additional Information: [full citation](#), [abstract](#), [references](#), [citations](#), [index terms](#)

A new algorithm for distributed garbage collection is presented. This algorithm collects distributed garbage incrementally and concurrently with user activity. It is the first incremental algorithm that is capable of collecting cyclic distributed garbage. Computational and network communication overhead are acceptable. Hosts may be temporarily inaccessible and synchronization between hosts is not necessary. The algorithm is based on asynchronous distribution of timestamp pa ...

12 An experimental study of renewal-older-first garbage collection



Lars T. Hansen, William D. Clinger

September 2002 **ACM SIGPLAN Notices , Proceedings of the seventh ACM SIGPLAN international conference on Functional programming ICFP '02**, Volume 37 Issue 9

Publisher: ACM Press

Full text available: [pdf\(143.87 KB\)](#) Additional Information: [full citation](#), [abstract](#), [references](#), [citations](#), [index terms](#)

Generational collection has improved the efficiency of garbage collection in fast-allocating programs by focusing on collecting young garbage, but has done little to reduce the cost of collecting a heap containing large amounts of older data. A new generational technique, older-first collection, shows promise in its ability to manage older data. This paper reports on an implementation study that compared two older-first collectors to traditional (younger-first) generational collectors. One of the ...

Keywords: generational garbage collection, older-first

13 Beltway: getting around garbage collection gridlock



Stephen M Blackburn, Richard Jones, Kathryn S. McKinley, J Eliot B Moss
May 2002 **ACM SIGPLAN Notices , Proceedings of the ACM SIGPLAN 2002 Conference
on Programming language design and implementation PLDI '02**, Volume 37
Issue 5

Publisher: ACM Press

Full text available: pdf(184.50 KB) . Additional Information: [full citation](#), [abstract](#), [references](#), [citings](#), [index terms](#)

We present the design and implementation of a new garbage collection framework that significantly generalizes existing copying collectors. The *Beltway* framework exploits and separates object age and incrementality. It groups objects in one or more increments on queues called *belts*, collects belts independently, and collects increments on a belt in first-in-first-out order. We show that Beltway configurations, selected by command line options, act and perform the same as semi-space, ...

Keywords: Java, beltway, copying collection, generational collection

14 Combining region inference and garbage collection



Niels Hallenberg, Martin Elsmann, Mads Tofte
May 2002 **ACM SIGPLAN Notices , Proceedings of the ACM SIGPLAN 2002 Conference
on Programming language design and implementation PLDI '02**, Volume 37
Issue 5

Publisher: ACM Press

Full text available: pdf(195.49 KB) . Additional Information: [full citation](#), [abstract](#), [references](#), [citings](#), [index terms](#)

This paper describes a memory discipline that combines region-based memory management and copying garbage collection by extending Cheney's copying garbage collection algorithm to work with regions. The paper presents empirical evidence that region inference very significantly reduces the number of garbage collections; and evidence that the fastest execution is obtained by using regions alone, without garbage collection. The memory discipline is implemented for Standard ML in the ML Kit compiler ...

Keywords: garbage collection, region interface, standard ML

15 Accurate garbage collection in an uncooperative environment



Fergus Henderson
June 2002 **ACM SIGPLAN Notices , Proceedings of the 3rd international symposium
on Memory management ISMM '02**, Volume 38 Issue 2 supplement

Publisher: ACM Press

Full text available: pdf(136.65 KB) . Additional Information: [full citation](#), [abstract](#), [references](#), [citings](#), [index terms](#)

Previous attempts at garbage collection in uncooperative environments have generally used conservative or mostly-conservative approaches. We describe a technique for doing fully type-accurate garbage collection in an uncooperative environment, using a "shadow stack" to link structs of pointer-containing variables, together with the data or code needed to trace them. We have implemented this in the Mercury compiler, which generates C code, and present preliminary performance data on the overheads ...

Keywords: C, garbage collection, multithreading, programming language implementation

16

Using passive object garbage collection algorithms for garbage collection of active



objects

Abhay Vardhan, Gul Agha

June 2002 **ACM SIGPLAN Notices , Proceedings of the 3rd international symposium on Memory management ISMM '02**, Volume 38 Issue 2 supplement

Publisher: ACM Press

Full text available: pdf(192.46 KB) Additional Information: [full citation](#), [abstract](#), [references](#), [index terms](#)

With the increasing use of active object systems, agents and concurrent object oriented languages like Java, the problem of garbage collection (GC) of unused resources has become more complex. Since active objects are autonomous computational agents, unlike passive object systems the criterion for identifying garbage in active objects cannot be based solely on reachability from a root set. This has led to development of specialized algorithms for GC of active objects. We reduce the problem of GC ...

Keywords: Java, active objects, actors, agents, garbage collection, program transformation

17 Error-free garbage collection traces: how to cheat and not get caught



Matthew Hertz, Stephen M Blackburn, J Eliot B Moss, Kathryn S. McKinley, Darko Stefanović

June 2002 **ACM SIGMETRICS Performance Evaluation Review , Proceedings of the 2002 ACM SIGMETRICS international conference on Measurement and modeling of computer systems SIGMETRICS '02**, Volume 30 Issue 1

Publisher: ACM Press

Full text available: pdf(105.06 KB) Additional Information: [full citation](#), [abstract](#), [references](#), [citations](#)

Programmers are writing a large and rapidly growing number of programs in object-oriented languages such as Java that require garbage collection (GC). To explore the design and evaluation of GC algorithms quickly, researchers are using simulation based on traces of object allocation and lifetime behavior. The *brute force* method generates perfect traces using a whole-heap GC at every potential GC point in the program. Because this process is prohibitively expensive, researchers often use < ...

18 Conservative garbage collection for general memory allocators



Gustavo Rodriguez-Rivera

October 2000 **ACM SIGPLAN Notices , Proceedings of the 2nd international symposium on Memory management ISMM '00**, Volume 36 Issue 1

Publisher: ACM Press

Full text available: pdf(829.20 KB) Additional Information: [full citation](#), [abstract](#), [citations](#), [index terms](#)

This paper explains a technique that integrates conservative garbage collection on top of general memory allocators. This is possible by using two data structures named malloc-tables and jump-tables that are computed at garbage collection time to map pointers to beginning of objects and their sizes. This paper describes malloc-tables and jump-tables, an implementation of a malloc/jump-table based conservative garbage collector for Doug Lea's memory allocator, and experimental results that com ...

Keywords: automatic memory management, conservative garbage collection, memory allocation

19 On the type accuracy of garbage collection



Martin Hirzel, Amer Diwan

October 2000 **ACM SIGPLAN Notices , Proceedings of the 2nd international symposium on Memory management ISMM '00**, Volume 36 Issue 1

Publisher: ACM Press

Full text available: pdf(1.25 MB) Additional Information: [full citation](#), [abstract](#), [citations](#), [index terms](#)

We describe a novel approach to obtaining type-accurate information for garbage collection in a hardware and language independent way. Our approach uses a run-time analysis to propagate pointer/non-pointer information from significant type events (such as allocation, which always returns a pointer). We use this technique to perform a detailed comparison of garbage collectors with different levels of accuracy and explicit deallocation on a range of C programs. We take advantage of the portabil ...

20 [Age-based garbage collection](#)



Darko Stefanović, Kathryn S. McKinley, J. Eliot B. Moss

October 1999 **ACM SIGPLAN Notices , Proceedings of the 14th ACM SIGPLAN conference on Object-oriented programming, systems, languages, and applications OOPSLA '99**, Volume 34 Issue 10

Publisher: ACM Press

Full text available: [pdf\(1.47 MB\)](#)

Additional Information: [full citation](#), [abstract](#), [references](#), [citations](#), [index terms](#)

Modern generational garbage collectors look for garbage among the young objects, because they have high mortality; however, these objects include the very youngest objects, which clearly are still live. We introduce new garbage collection algorithms, called age-based, some of which postpone consideration of the youngest objects. Collecting less than the whole heap requires write barrier mechanisms to track pointers into the collected region. We describe her ...

Keywords: garbage collection, generational and copy collection, object behavior, write barrier

Results 1 - 20 of 200

Result page: [1](#) [2](#) [3](#) [4](#) [5](#) [6](#) [7](#) [8](#) [9](#) [10](#) [next](#)

The ACM Portal is published by the Association for Computing Machinery. Copyright © 2006 ACM, Inc.

[Terms of Usage](#) [Privacy Policy](#) [Code of Ethics](#) [Contact Us](#)

Useful downloads: [Adobe Acrobat](#) [QuickTime](#) [Windows Media Player](#) [Real Player](#)